INSTRUCTION MANUAL

# GPS16X-HVS GPS Receiver

Revision: 9/15

Copyright © 2003-2015
Campbell Scientific, Inc.

# *Limited Warranty*

"Products manufactured by CSI are warranted by CSI to be free from defects in materials and workmanship under normal use and service for twelve months from the date of shipment unless otherwise specified in the corresponding product manual. (Product manuals are available for review online at *www.campbellsci.com*.) Products not manufactured by CSI, but that are resold by CSI, are warranted only to the limits extended by the original manufacturer. Batteries, fine-wire thermocouples, desiccant, and other consumables have no warranty. CSI's obligation under this warranty is limited to repairing or replacing (at CSI's option) defective Products, which shall be the sole and exclusive remedy under this warranty. The Customer assumes all costs of removing, reinstalling, and shipping defective Products to CSI. CSI will return such Products by surface carrier prepaid within the continental United States of America. To all other locations, CSI will return such Products best way CIP (port of entry) per Incoterms ® 2010. This warranty shall not apply to any Products which have been subjected to modification, misuse, neglect, improper service, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied. The warranty for installation services performed by CSI such as programming to customer specifications, electrical connections to Products manufactured by CSI, and Product specific training, is part of CSI's product warranty. **CSI EXPRESSLY DISCLAIMS AND EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE**. **CSI hereby disclaims, to the fullest extent allowed by applicable law, any and all warranties and conditions with respect to the Products, whether express, implied or statutory, other than those expressly provided herein.**"

# Assistance

Products may not be returned without prior authorization. The following contact information is for US and international customers residing in countries served by Campbell Scientific, Inc. directly. Affiliate companies handle repairs for customers within their territories. Please visit *www.campbellsci.com* to determine which Campbell Scientific company serves your country.

To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (435) 227-9000. After an application engineer determines the nature of the problem, an RMA number will be issued. Please write this number clearly on the outside of the shipping container. Campbell Scientific's shipping address is:

**CAMPBELL SCIENTIFIC, INC.**
RMA#_____
815 West 1800 North
Logan, Utah 84321-1784

For all returns, the customer must fill out a "Statement of Product Cleanliness and Decontamination" form and comply with the requirements specified in it. The form is available from our web site at *www.campbellsci.com/repair*. A completed form must be either emailed to *repair@campbellsci.com* or faxed to (435) 227-9106. Campbell Scientific is unable to process any returns until we receive this form. If the form is not received within three days of product receipt or is incomplete, the product will be returned to the customer at the customer's expense. Campbell Scientific reserves the right to refuse service on products that were exposed to contaminants that may cause health or safety concerns for our employees.

# Safety

DANGER — MANY HAZARDS ARE ASSOCIATED WITH INSTALLING, USING, MAINTAINING, AND WORKING ON OR AROUND **TRIPODS, TOWERS, AND ANY ATTACHMENTS TO TRIPODS AND TOWERS SUCH AS SENSORS, CROSSARMS, ENCLOSURES, ANTENNAS, ETC**. FAILURE TO PROPERLY AND COMPLETELY ASSEMBLE, INSTALL, OPERATE, USE, AND MAINTAIN TRIPODS, TOWERS, AND ATTACHMENTS, AND FAILURE TO HEED WARNINGS, INCREASES THE RISK OF DEATH, ACCIDENT, SERIOUS INJURY, PROPERTY DAMAGE, AND PRODUCT FAILURE. TAKE ALL REASONABLE PRECAUTIONS TO AVOID THESE HAZARDS. CHECK WITH YOUR ORGANIZATION'S SAFETY COORDINATOR (OR POLICY) FOR PROCEDURES AND REQUIRED PROTECTIVE EQUIPMENT PRIOR TO PERFORMING ANY WORK.

Use tripods, towers, and attachments to tripods and towers only for purposes for which they are designed. Do not exceed design limits. Be familiar and comply with all instructions provided in product manuals. Manuals are available at www.campbellsci.com or by telephoning (435) 227-9000 (USA). You are responsible for conformance with governing codes and regulations, including safety regulations, and the integrity and location of structures or land to which towers, tripods, and any attachments are attached. Installation sites should be evaluated and approved by a qualified engineer. If questions or concerns arise regarding installation, use, or maintenance of tripods, towers, attachments, or electrical connections, consult with a licensed and qualified engineer or electrician.

General

- Prior to performing site or installation work, obtain required approvals and permits. Comply with all governing structure-height regulations, such as those of the FAA in the USA.
- Use only qualified personnel for installation, use, and maintenance of tripods and towers, and any attachments to tripods and towers. The use of licensed and qualified contractors is highly recommended.
- Read all applicable instructions carefully and understand procedures thoroughly before beginning work.
- Wear a **hardhat** and **eye protection**, and take **other appropriate safety precautions** while working on or around tripods and towers.
- **Do not climb** tripods or towers at any time, and prohibit climbing by other persons. Take reasonable precautions to secure tripod and tower sites from trespassers.
- Use only manufacturer recommended parts, materials, and tools.

Utility and Electrical

- **You can be killed** or sustain serious bodily injury if the tripod, tower, or attachments you are installing, constructing, using, or maintaining, or a tool, stake, or anchor, come in **contact with overhead or underground utility lines**.
- Maintain a distance of at least one-and-one-half times structure height, 20 feet, or the distance required by applicable law, **whichever is greater**, between overhead utility lines and the structure (tripod, tower, attachments, or tools).
- Prior to performing site or installation work, inform all utility companies and have all underground utilities marked.
- Comply with all electrical codes. Electrical equipment and related grounding devices should be installed by a licensed and qualified electrician.

Elevated Work and Weather

- Exercise extreme caution when performing elevated work.
- Use appropriate equipment and safety practices.
- During installation and maintenance, keep tower and tripod sites clear of un-trained or non-essential personnel. Take precautions to prevent elevated tools and objects from dropping.
- Do not perform any work in inclement weather, including wind, rain, snow, lightning, etc.

Maintenance

- Periodically (at least yearly) check for wear and damage, including corrosion, stress cracks, frayed cables, loose cable clamps, cable tightness, etc. and take necessary corrective actions.
- Periodically (at least yearly) check electrical ground connections.

WHILE EVERY ATTEMPT IS MADE TO EMBODY THE HIGHEST DEGREE OF SAFETY IN ALL CAMPBELL SCIENTIFIC PRODUCTS, THE CUSTOMER ASSUMES ALL RISK FROM ANY INJURY RESULTING FROM IMPROPER INSTALLATION, USE, OR MAINTENANCE OF TRIPODS, TOWERS, OR ATTACHMENTS TO TRIPODS AND TOWERS SUCH AS SENSORS, CROSSARMS, ENCLOSURES, ANTENNAS, ETC.

# Table of Contents

*PDF viewers: These page numbers refer to the printed version of this document. Use the PDF reader bookmarks tab for links to specific sections.*


**1. Overview ..... 1**

1.1 Default Settings ..... 1
1.2 Compatible Dataloggers ..... 2
1.3 Common Accessories ..... 2

**2. Specifications ..... 3**

**3. Wiring ..... 4**

3.1 Using with an A300 ..... 5

**4. GPS Data ..... 6**

4.1 $GPGGA Sentence (Position and Time) ..... 6
4.2 $GPRMC Sentence (Position and Time) ..... 7

**5. CRBasic Programming ..... 8**

5.1 GPS() Instruction ..... 8
5.2 Example Program Using GPS() Instruction ..... 10
5.3 Example Program Using Serial Instruction ..... 11

**6. Troubleshooting ..... 13**

6.1 Testing and Evaluating Serial Communications ..... 13
6.2 NMEAStrings Variable Populated, but Clock Not Setting ..... 14

***Appendices***

**A. Changing GPS16X-HVS Settings ..... A-1**

A.1 Computer Connections ..... A-1
A.1.1 Using the A200 ..... A-1
A.1.1.1 Driver Installation ..... A-1
A.1.1.2 Wiring ..... A-2
A.1.1.3 Powering the Sensor ..... A-3
A.1.1.4 Determining which COM Port the A200 has been Assigned ..... A-3
A.1.2 Using the SC110 ..... A-3

**B. CR23X Wiring and Programming ..... B-1**

B.1 CR23X Connections ..... B-1
B.2 Programming ..... B-1
B.2.1 Program Execution Interval ..... B-1
B.2.2 Reading GPS Data ..... B-2
B.2.2.1 CR23X Example of Instruction 15 (P15) ..... B-3

B.2.3 Filters ..... B-3
B.2.4 Managing the Data ..... B-3
B.2.5 Program Discussion ..... B-4
B.2.5.1 CR23X Example Program ..... B-6
B.3 Troubleshooting ..... B-8

**C. Using an SDM-SIO4 ..... C-1**

C.1 SDM-SIO4 Connections ..... C-1
C.2 CR9000X Example Program ..... C-1

## *Figures*

1-1. The GPS16X-HVS terminates in pigtails for direct connection to our dataloggers ..... 1
3-1. CR1000 to GPS16X-HVS connection ..... 5
A-1. A200 Sensor-to-PC Interface ..... A-2

## *Tables*

1-1. Default Settings ..... 1
3-1. Datalogger Wiring ..... 4
5-1. GPS16X-HVS Wiring to A300 Terminals and Datalogger Terminals ..... 5
5-2. A300 Cable Wiring to Datalogger Terminals ..... 6
4-1. NMEA $GPGGA String Definition ..... 6

A-1. A200 Wiring ..... A-2
A-2. SC110's DCE Cable Wiring ..... A-3
B-1. CR23X Wiring ..... B-1
B-2. P15 for NMEA $GPGGA Data String ..... B-2
B-3. Filter ..... B-3
C-1. SC110's Cable Wiring ..... C-1

# GPS16X-HVS GPS Receiver

## 1. Overview

*FIGURE 1-1. The GPS16X-HVS terminates in pigtails for direct connection to our dataloggers*

The GPS16X-HVS is a complete GPS receiver manufactured by Garmin International, Inc. Campbell Scientific configures the GPS16X-HVS to work with our dataloggers and modifies its cable so that the cable terminates in pigtails. The pigtails connect directly to the control ports of our dataloggers or with the aid of an A300.

The GPS16X-HVS includes the GPS receiver and antenna in the same housing with one cable for the power supply and communications. The GPS antenna must have a clear view of the sky. Generally the GPS antenna will not work indoors.

The GPS16X-HVS is a 12-channel GPS receiver that supports FAA Wide Area Augmentation System (WAAS) or RTCM differential GPS. Also supported is the 1 Pulse Per Second (PPS) timing signal. The cable connections provided with the GPS16X-HVS do not support differential GPS correction. The cable can be modified by the user if differential correction is required.

### 1.1 Default Settings

TABLE 1-1 shows the default settings of the GPSX16-HVS.

**TABLE 1-1. Default Settings**

| | |
|---|---|
| Baud Rate | 38400 bps |
| Parity | N (no parity) |
| Stop Bit | 1 |
| Sentences Output | GPGGA, GPRMC |
| PPS | 100 ms |

## 1.2 Compatible Dataloggers

**Compatible Contemporary Dataloggers**

| CR200(X) Series | CR800/ CR850 | CR1000 | CR3000 | CR6 | CR9000X |
|---|---|---|---|---|---|
| | ✓* | ✓* | ✓* | ✓ | See Appendix B and C |

*If PPS is required, the A300 Power and Signal Converter is needed.

**Compatible Retired Dataloggers**

| CR500 | CR510 | CR10 | CR10X | 21X | CR23X | CR9000 | CR5000 | CR7X |
|---|---|---|---|---|---|---|---|---|
| | | | | | See Appendix B and C | See Appendix B and C | See Appendix B and C | |

Our CR6, CR800, CR850, CR1000, and CR3000 dataloggers typically use the CRBasic **GPS()** instruction to read the GPS16X-HVS. To use the PPS functionality, some dataloggers need an updated clock chip. The clock chip is factory replaced (requires an RMA). Dataloggers with the following serial numbers need an updated chip:

| Datalogger | Serial Number |
|---|---|
| CR1000M | < 20409 |
| CR800, CR850 | < 7920 |
| CR3000 | < 3168 |

In August 2014, Garmin changed the GPS16X-HVS PPS output signal from 5 V to 3 V. Units with serial numbers greater than 1A4189318 have a 3 V PPS output signal. When this new design is used with a CR800, CR850, CR1000, or CR3000 datalogger, a 3 V to 5 V voltage shifter is required for use with the PPS signal output. The A300 can be used for this purpose. This level shifter is NOT required for the CR6 datalogger.

## 1.3 Common Accessories

| CSI part number | Description |
|---|---|
| 17212 | GPS16X-HVS magnetic mount |
| CM235 | Magnetic mounting stand |
| A200 | Sensor to PC interface |

# 2. Specifications

**Physical**

**Size:** 86 mm (3.39 in) diameter, 42 mm (1.65 in) high

**Weight:** 181 g (6.4 oz) without cable, 332 g (11.7 oz) with 5 m cable

**Cable:** PVC-jacketed, 5 m, foil-shielded, 8-conductor, 28 AWG

**Electrical Characteristics**

**Input Voltage:** 8.0 Vdc to 40 Vdc unregulated

**Current Drain:** 65 mA @ 12 Vdc

**GPS Receiver**

**Sensitivity:** –185 dbW minimum

**GPS Performance**

**Receiver:** WAAS enabled; 12 parallel channel GPS receiver continuously tracks and uses up to 12 satellites, 11 if PPS is active

**Acquisition Times (Approximate)**

- **Reacquisition:** Less than 2 s
- **Hot:** 1 s (all data known)
- **Warm:** ~38 s (initial position, time and almanac known, ephemeris unknown)
- **Cold:** ~45 s

**SkySearch:** 5 min (no data known)

**Sentence Rate:** 1 s default; NMEA 0183 output interval configurable from 1 to 900 s in one second increments

**Accuracy:** GPS Standard Positioning Service (SPS)

- **Position:** Less than 15 m, 95% typical (100 m with selective availability on)
- **Velocity:** 0.1 knot RMS steady state

**DGPS (USCG/RTCM)**

- **Position:** 3-5 m, 95% typical
- **Velocity:** 0.1 knot RMS steady state

**DGPS (WAAS)**

- **Position:** Less than 3 m
- **Velocity:** 0.1 knot RMS steady state

**PPS Time:** ±1 microsecond at rising edge of PPS pulse (subject to selective availability)

**Dynamics:** 999 knots velocity (limited above 60,000 ft, 6g dynamics)

**Interfaces:** True RS-232 output, asynchronous serial input compatible with RS-232 or TTL voltage levels, RS-232 polarity. Selectable baud rates (4800, 9600, 19200, 38400)

**PPS:** 1 Hz pulse, programmable width, 1 microsecond accuracy

**Power Control**
**Off:** Open circuit
**On:** Ground or pull to low logic level < 0.3 volts

**Environmental Characteristics**

**Temperature:** –30 to 80 °C operational, –40 to 80 °C storage

# 3. Wiring

The GPS16X-HVS connects directly to a CR6, CR800, CR850, CR1000, or CR3000 datalogger (see TABLE 3-1). However, if PPS is required, the A300 Power and Signal Converter may be required for use with the CR800/850, CR1000, and CR3000. See Section 3.1, *Using with an A300* (p. 5). The CR6 does not require the use of an A300.

Refer to Appendix B.1, *CR23X Connections* (p. B-1), if connecting the receiver to a CR23X. Our CR5000 and CR9000X connect to the receiver via the SC110 and an SDM-SIO4 (see Appendix C, *Using an SDM-SIO4* (p. C-1)).

If the GPS16X-HVS is to be connected to a computer to change the default settings, an A200 or SC110 cable is needed (see Appendix A, *Changing GPS16X-HVS Settings* (p. A-1)).

**TABLE 3-1. Datalogger Wiring**

| GPS16X-HVS | Datalogger | Function |
|---|---|---|
| Red | 12V | Power In |
| Black | Ground | Power Ground |
| Yellow | Ground or Control Port for On/Off control | Power Switch |
| White | Control Port (Rx) | TXD |
| Gray | Control Port (Tx) | PPS |
| Blue | Ground or Control Port (Tx) for datalogger-based configuration | $R_x$ data |
| Shield | Ground | Shield |

*FIGURE 3-1. CR1000 to GPS16X-HVS connection*

## 3.1 Using with an A300

In 2014, Garmin changed the pulse-per-second (PPS) output of the GPS16X-HVS from 5 V to 3 V. Units with a serial number 1A4189318 or greater have a PPS output of 0 to 3 V. For those units, an A300 is needed to connect the PPS output to a CR800-series, CR1000, or CR3000 datalogger. Those dataloggers require the PPS line to have a voltage of 3.8 V or greater.

**TABLE 3-2. GPS16X-HVS Wiring to A300 Terminals and Datalogger Terminals**

| GPS16X-HVS Wire Color | GPS16X-HVS Wire Function | A300 Terminal | Datalogger |
|---|---|---|---|
| Red | 12 V | | 12V |
| Black | Ground | G | |
| Yellow | Enable | | Ground (or Control Port) |
| White | TXD (Output) | | Control Port (Rx) |
| Gray | PPS | 3.3V IN | |
| Blue | RXD (Input) | | Ground |
| Shield | Shield | | Ground |

| TABLE 3-3. A300 Cable Wiring to Datalogger Terminals | | |
|---|---|---|
| **A300 Wire Color** | **A300 Wire Function** | **Datalogger** |
| Red | 12 V | 12V |
| Black | Ground | Ground |
| Green | 5 V Signal Input | Ground |
| White | 5 V Signal Output | Control Port (Tx) |

# 4. GPS Data

The GPS16X-HVS has several data formats available. The GPS16X-HVS is configured to output the NMEA $GPGGA and $GPRMC time and position string. It is possible to configure the GPS16X-HVS to output other NMEA strings including the $GPVTG track made good and ground speed string. See Appendix A, *Changing GPS16X-HVS Settings* *(p. A-1)*, for details.

## 4.1 $GPGGA Sentence (Position and Time)

Sample NMEA $GPGGA data string:

$GPGGA,hhmmss,llll.lll,a,nnnnn.nnn,b,t,uu,v.v,w.w,M,x.x,M,y.y,zzzz*hh<CR><LF>

| TABLE 4-1. NMEA $GPGGA String Definition | | |
|---|---|---|
| **Field** | **Description** | |
| 0 | $GPGGA | NMEA string identifier |
| 1 | hhmmss | UTC of Position: Hours, minutes, seconds |
| 2 | 1111.111 | Latitude: Degrees, minutes, thousandths of minutes |
| 3 | a | N (North) or S (South) |
| 4 | nnnnn.nnn | Longitude: Degrees, minutes, thousandths of minutes |
| 5 | b | E (East) or W (West) |
| 6 | t | GPS Quality Indicator: 0 = No GPS, 1 = GPS, 2 = DGPS |
| 7 | uu | Number of Satellites in Use |
| 8 | v.v | Horizontal Dilution of Precision (HDOP) |
| 9 | w.w | Antenna Altitude in Meters |
| 10 | M | M = Meters |
| 11 | x.x | Geoidal Separation in Meters |
| 12 | M | M = Meters. Geoidal separation is the difference between the WGS-84 earth ellipsoid and mean-sea-level. |
| 13 | y.y | Age of Differential GPS Data. Time in seconds since the last Type 1 or 9 Update |
| 14 | zzzz | Differential Reference Station ID (0000 to 1023) |
| 15 | * | Asterisk, generally used as the termination character |
| 16 | hh | Checksum |
| 17 | <CR><LF> | Carriage return, line feed characters. |

Sample $GPGGA output strings:

Cold Start
No satellites acquired, Real Time Clock and Almanac invalid:
$GPGGA,,,,,,0,00,,,,,,,*66

Warm Start
No satellites acquired, time from Real Time Clock, almanac valid:
$GPGGA,235032.0,,,,,0,00,,,,,,,*7D

Warm Start
One satellite in use, time from GPS Real Time Clock (not GPS), no position:
$GPGGA,183806.0,,,,,0,01,,,,,,,*7D

Valid GPS Fix
Three satellites acquired, time and position valid:
$GPGGA,005322.0,4147.603,N,11150.978,W,1,03,11.9,00016,M,-016,M,,*6E

## 4.2 $GPRMC Sentence (Position and Time)

Example (signal not acquired):
$GPRMC,235947.000,V,0000.0000,N,00000.0000,E,,,041299,,*1D
Example (signal acquired):
$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,*25

| Field | Example | Comments |
|---|---|---|
| Sentence ID | $GPRMC | |
| UTC Time | 092204.999 | hhmmss.sss |
| Status | A | A = Valid, V = Invalid |
| Latitude | 4250.5589 | ddmm.mmmm |
| N/S Indicator | S | N = North, S = South |
| Longitude | 14718.5084 | dddmm.mmmm |
| E/W Indicator | E | E = East, W = West |
| Speed over ground | 0.00 | Knots |
| Course over ground | 0.00 | Degrees |
| UTC Date | 211200 | DDMMYY |
| Magnetic variation | | Degrees |
| Magnetic variation | | E = East, W = West |
| Checksum | *25 | |
| Terminator | CR/LF | |

# 5. CRBasic Programming

This section describes programming a CR6, CR800, CR850, CR1000, or CR3000. See Appendix B, *CR23X Wiring and Programming* (p. B-1), and Appendix C, *Using an SDM-SIO4* (p. C-1), for programming other dataloggers.

CRBasic is used to write programs for the CR6, CR1000, CR3000, CR800, and CR850 dataloggers. These dataloggers use several instructions to read GPS output, which is asynchronous serial data.

## 5.1 GPS() Instruction

The **GPS()** instruction is available for our CR6, CR800, CR850, CR1000, and CR3000 dataloggers. It is used along with a GPS device to set the datalogger's clock. This instruction will also provide information such as location (latitude/longitude) and speed, and store NMEA sentences from the GPS device.

**NOTE** To use the **GPS()** instruction, the datalogger operating system (OS) should be OS17 or higher for the CR1000; OS10 or higher for the CR3000; or OS08 or higher for the CR800 and CR850. Go to *www.campbellsci.com/downloads* to upgrade the datalogger OS.

The resolution of accuracy for the clock set is 10 microseconds if the datalogger has a hardware revision number greater than 007 (RevBoard field in the datalogger's Status table). Otherwise, resolution is 10 milliseconds. The clock set relies on information from the GPRMC sentence. If this sentence is not returned, a clock set will not occur.

By default, the instruction expects the GPS unit to be set up at 38400 baud, outputting the GPRMC and GPGGA sentences once per second. The datalogger expects the start of the second to coincide with the rising edge of the PPS signal. If there is no PPS signal or if the required sentences come out at less than once per second, the datalogger will not update its clock.

GPS units with lower baud rates can be used with the **GPS()** instruction but the baud rate has to be set for the relevant Com port it is to be connected to either in the datalogger settings or by including a **SetStatus()** command after the **BeginProg()** instruction in the program (e.g., **SetStatus**(*"BaudrateCOM4",19200*)).

Baud rates of 2400 bps or lower will not work as the GPS unit will not transmit the two GPS sentences once per second reliably. Similar problems can be encountered even at higher baud rates if too many optional GPS strings are selected to be output.

The **GPS()** instruction has the following syntax:

**GPS**(*GPSArray,ComPort,TimeOffset,MaxTimeDiff,NMEAStrings*)

Description of the parameters follows:

*GPSArray* The *GPSArray* parameter is the variable in which to store the information returned by the GPS. Fifteen values are returned. If this array is not dimensioned to 15, values will be stored to fill the array and no error will be returned. If no values are available, NAN will be returned. The following values are returned by the GPS:

Array(1) = Latitude, degrees
Array(2) = Latitude, minutes
Array(3) = Longitude, degrees
Array(4) = Longitude, minutes
Array(5) = Speed over ground, knots
Array(6) = Course over ground, degrees
Array(7) = Magnetic variation (positive = East, negative = West)
Array(8) = Fix Quality (0 = invalid, 1 = GPS, 2 = differential GPS, 6 = estimated)
Array(9) = Number of Satellites
Array(10) = Altitude, meters
Array(11) = Pulse per second (PPS) length, microseconds
Array(12) = Seconds since last GPRMC sentence
Array(13) = GPS Ready, 10 = ready
Array(14) = Maximum clock change, milliseconds (10 msec resolution)
Array(15) = Clock change count

*ComPort* The *ComPort* parameter is the control port pair to which the GPS device is attached. Valid options are COM1 (C1/C2), COM2 (C3/C4), COM3 (C5/C6), and COM4 (C7/C8). Rx is used to read in the NMEA sentences and Tx is used to monitor the PPS from the GPS. This instruction defaults to a baud rate of 38,400 bps. If a different baud rate is required, use the **SetStatus()** instruction to override the default.

*TimeOffset* The *TimeOffset* parameter is the local time offset, in seconds, from UTC.

*MaxTimeDiff* The *MaxTimeDiff* parameter is the maximum difference in time between the datalogger clock and the GPS clock that will be tolerated before the clock is changed. If a negative value is entered, the clock will not be changed.

For dataloggers prior to hardware revision 08, the *MaxTimeDiff* parameter should not be set to 0. A minimum value of 20 ms is recommended. With this hardware, when a **GPS()** instruction is in the program the clock is checked each second (regardless of how often the **GPS()** instruction is run). The clock is set if any difference is found. This can result in the clock being set each second, resulting in skipped records in the data table(s). This restriction does not apply to hardware revisions 08 or greater.

*NMEAStrings* The *NMEAStrings* parameter is the string array that holds the NMEA sentences. If it exists, the GPRMC sentence will reside in NMEAStrings(1), and the GPGGA sentence will reside in NMEAStrings(2). Any other sentences will reside in subsequent indexes into the array (on a first-in basis). Once an index in the array is used to store a particular sentence, that sentence will always be stored in that location when updates to the sentence are received.

## 5.2 Example Program Using GPS() Instruction

The following wiring and short program provide an example of using the **GPS()** instruction with the Garmin GPS16X-HVS.

```
'Program the GPS16-HVS to use 38.4 kbaud, no parity, 8 data bits, and 1 stop bit
PipeLineMode

Const LOCAL_TIME_OFFSET = -6            'Local time offset relative to UTC time
Dim nmea_sentence(2) As String * 90

Public gps_data(15)
Alias gps_data(1) = latitude_a          'Degrees latitude (+ = North; - = South)
Alias gps_data(2) = latitude_b          'Minutes latitude
Alias gps_data(3) = longitude_a         'Degress longitude (+ = East; - = West)
Alias gps_data(4) = longitude_b         'Minutes longitude
Alias gps_data(5) = speed               'Speed
Alias gps_data(6) = course              'Course over ground
Alias gps_data(7) = magnetic_variation  'Magnetic variation from true north (+ =
                                        'East; - = West)
Alias gps_data(8) = fix_quality         'GPS fix quality: 0 = invalid, 1 = GPS, 2 =
                                        'differential GPS, 6 = estimated
Alias gps_data(9) = nmbr_satellites     'Number of satellites used for fix
Alias gps_data(10) = altitude           'Antenna altitude
Alias gps_data(11) = pps                'usec into sec of system clock when PPS
                                        'rising edge occurs, typically 990,000 once
                                        'synced
Alias gps_data(12) = dt_since_gprmc     'Time since last GPRMC string, normally less
                                        'than 1 second
Alias gps_data(13) = gps_ready          'Counts from 0 to 10, 10 = ready
Alias gps_data(14) = max_clock_change   'Maximum value the clock was changed in msec
Alias gps_data(15) = nmbr_clock_change  'Number of times the clock was changed

'Define Units to be used in data file header
Units latitude_a = degrees
Units latitude_b = minutes
Units longitude_a = degrees
Units longitude_b = minutes
Units speed = knots
Units course = degrees
Units magnetic_variation = unitless
Units fix_quality = unitless
Units nmbr_satellites = unitless
Units altitude = m
Units pps = ms
Units dt_since_gprmc = s
Units gps_ready = unitless
```

```
Units max_clock_change = ms
Units nmbr_clock_change = samples

BeginProg
    'Use SetStatus prior to scan if baud rate needs to be changed for device
    Scan (1,Sec,0,0)
      GPS (latitude_a,Com4,LOCAL_TIME_OFFSET*3600,100,nmea_sentence(1))
    NextScan
EndProg
```

## 5.3 Example Program Using Serial Instruction

Serial programming allows the retrieval of all values of GPRMC and GPGGA values. The **GPS()** instruction is a subset of the values that are available.

```
'GPS16X-HVS at Campbell Scientific Factory Defaults
Const GPSPort = Com4 'Com port where GPS is connected

Public GGAstring As String * 500
Public RMCstring As String * 500
'rmc variables
Public rmcid As String
Public rmcutc As String
Public rmcstatus As String
Public rmclatitude As String
Public rmcin_s_ind As String
Public rmclongitude As String
Public rmce_w_indicator As String
Public rmcspeed As String
Public rmccourse As String
Public rmcutcdate As String
Public rmcmagvariation As String
Public rmcmage_w As String
Public rmccchecksum As String
'gga variables
Public ggaid As String
Public ggautc As String
Public ggailatitude As String
Public ggan_s_ind As String
Public ggalongitude As String
Public ggae_w_ind As String
Public ggapositionfix As String
Public gganumsatellites As String
Public ggahdop As String
Public ggaaltitude As String
Public ggaaltutudeunits As String
Public ggageoidsep As String
Public ggageoidunits As String
Public ggachecksum As String

Dim NBytesReturned As Long
Dim SubStrings(16) As String * 32, rawdata As String * 500
Dim CalculatedChecksum As Long, ReportedChecksum As Long
```

```
DataTable (gpsdata,True,-1)
  DataInterval (0,1,Sec,10)
  Sample (1,rmcid,String)
  Sample (1,rmcutc,String)
  Sample (1,rmcstatus,String)
  Sample (1,rmclatitude,String)
  Sample (1,rmcin_s_ind,String)
  Sample (1,rmclongitude,String)
  Sample (1,rmcspeed,String)
  Sample (1,rmccourse,String)
  Sample (1,rmcutcdate,String)
  Sample (1,rmcmagvariation,String)
  Sample (1,rmcmage_w,String)
  Sample (1,rmcchecksum,String)
  Sample (1,ggaid,String)
  Sample (1,ggautc,String)
  Sample (1,ggan_s_ind,String)
  Sample (1,ggalongitude,String)
  Sample (1,ggae_w_ind,String)
  Sample (1,ggapositionfix,String)
  Sample (1,gganumsatellites,String)
  Sample (1,ggahdop,String)
  Sample (1,ggaaltitude,String)
  Sample (1,ggaaltutudeunits,String)
  Sample (1,ggageoidsep,String)
  Sample (1,ggageoidunits,String)
  Sample (1,ggachecksum,String)
EndTable
'Main Program
BeginProg
  SerialOpen (GPSPort,38400,3,0,1001)
  Scan (1,Sec,0,0)

    SerialInRecord (GPSPort,rawdata,36,0,&h0D0A,NBytesReturned,11)
    CalculatedChecksum = CheckSum (rawdata,9,Len(rawdata) - 3)
    CalculatedChecksum = CalculatedChecksum AND 255
    ReportedChecksum = HexToDec(Right(rawdata,2))
    If CalculatedChecksum = ReportedChecksum Then
      If InStr (1,rawdata,"GPRMC",2) Then
        RMCstring = rawdata
      ElseIf InStr (1,rawdata,"GPGGA",2) Then
        GGAstring = rawdata
      EndIf
    EndIf
    SerialInRecord (GPSPort,rawdata,36,0,&h0D0A,NBytesReturned,11)
    CalculatedChecksum = CheckSum (rawdata,9,Len(rawdata) - 3)
    CalculatedChecksum = CalculatedChecksum AND 255
    ReportedChecksum = HexToDec(Right(rawdata,2))
    If CalculatedChecksum = ReportedChecksum Then
      If InStr (1,rawdata,"GPRMC",2) Then
        RMCstring = rawdata
      ElseIf InStr (1,rawdata,"GPGGA",2) Then
        GGAstring = rawdata
      EndIf
    EndIf
```

```
    'parse rmc data
    SplitStr (SubStrings(),RMCstring,",",16,5)
    rmcid = SubStrings(1)
    rmcutc = SubStrings(2)
    rmcstatus = SubStrings(3)
    rmclatitude = SubStrings(4)
    rmcin_s_ind =SubStrings(5)
    rmclongitude=SubStrings(6)
    rmce_w_indicator=SubStrings(7)
    rmcspeed=SubStrings(8)
    rmccourse=SubStrings(9)
    rmcutcdate=SubStrings(10)
    rmcmagvariation=SubStrings(11)
    rmcmage_w =Left(SubStrings(12),1)
    rmccheckSum=Right(RMCstring,2)

    'parse gga data
    SplitStr (SubStrings(),GGAstring,",",16,5)
    ggaid=SubStrings(1)
    ggautc=SubStrings(2)
    ggailatitude=SubStrings(3)
    ggan_s_ind=SubStrings(4)
    ggalongitude=SubStrings(5)
    ggae_w_ind=SubStrings(6)
    ggapositionfix=SubStrings(7)
    gganumsatellites=SubStrings(8)
    ggahdop=SubStrings(9)
    ggaaltitude=SubStrings(10)
    ggaaltutudeunits=SubStrings(11)
    ggageoidsep=SubStrings(12)
    ggageoidunits=Left(SubStrings(13),1)
    ggachecksum=Right(GGAstring,2)

    CallTable gpsdata
  NextScan
EndProg
```

# 6. Troubleshooting

Testing and evaluation of serial communications is best done by reducing the whole system to small manageable systems. Usually some portions of the whole system are working. The first steps involve finding what is working. During this process you may find parts of the system that are not working or mistakes that can be easily corrected. Fix each subsystem before testing others.

## 6.1 Testing and Evaluating Serial Communications

Test the GPS16X-HVS for proper operation including the baud rate and output string. Use a computer, terminal emulator software, a serial port (RS-232), and a 9-pin to pigtail cable (SC110/sockets). The computer and serial port can be the same as used to communicate with the datalogger. Terminal emulation software is common. Hyperterm is supplied as part of Windows™ and works. Procomm™ is another communication software package that works well.

Set up the software for the correct serial port, 38.4 kbps, 8 data bits, 1 stop bit and no parity. Flow control should be none. Using the SC110 cable, connect the GPS16X-HVS to the computer serial port. Power up the GPS16X-HVS. The GPS antenna should have a clear view of the sky. Don't expect the GPS antenna to work indoors. The $GPGGA and GPRMC strings should be displayed once a second. Make sure the $GPGGA string is showing a valid GPS fix. A valid GPS fix will display time, position and have a GPS quality number greater than zero.

| **SC110 Cable Connections** | |
|---|---|
| **GPS16X Receiver** | **SC110/Sockets or DB9/Sockets to Pigtails** |
| White | Pin 2 |
| Black and Yellow | Pin 5 (shares power ground) |

## 6.2 NMEAStrings Variable Populated, but Clock Not Setting

Look at the GPSReady variable. It will increment from 0 to 10 when the datalogger has received good GPRMC strings and a synchronized PPS signal. Once GPSReady reaches 10, the datalogger will begin to use GPS time for clock setting. The 12th value populated in GPSArray indicates elapsed time since a GPRMC string was received and should not exceed 1. If the GPRMC string is being received and GPSReady remains at zero, the PPS signal is not being received by the datalogger.

# *Appendix A. Changing GPS16X-HVS Settings*

As configured by Campbell Scientific, the GPS16X-HVS will output the NMEA 0183 $GPGGA and $GPRMC data strings once a second, the PPS signal is enabled with a duration of 100 milliseconds and the baud rate is set to 38,400 baud.

Special software (SNRSRCFG.EXE) is available from Garmin International for system setup. The GPS16X-HVS user manual available from Garmin International provides technical details beyond the scope of the Campbell Scientific user manual.

Settings used by Campbell Scientific for GPS16X-HVS setup:
GPS Base Model = GPS 16(X)
Fix Mode = Automatic
Baud Rate = 38,400
Dead Reckon Time = 30 sec
NMEA output time = 1 sec
Position pinning = off
NMEA 2.30 mode = off
Power Save Mode = off (Normal mode)
PPS mode = 1 Hz
PPS Length = 100 mS
Phaze output Data = off
DGPS Mode = WAAS only
Differential mode = Automatic
Earth Datum Index = WGS 84

Selected Sentences = GPGGA and GPRMC

Common changes would be baud rate and selected sentences. The NMEA 0183 GPVTG data sentence gives ground speed and direction, which may be required for some applications. Changes can be made with the Garmin software, or with a terminal emulator and the Garmin technical user manual. Contact Garmin International (www.garmin.com) for either resource.

## A.1 Computer Connections

Either an A200 interface or SC110 cable is required to connect the GPS16X-HVS to a computer. The A200 is used to connect to a computer USB port, and the SC110 is used to connect to a computer 9-pin serial port.

### A.1.1 Using the A200

#### A.1.1.1 Driver Installation

If the A200 has not been previously plugged into your PC and your PC operating system is not Windows 7, the A200 driver needs to be loaded onto your PC.

**NOTE** Drivers should be loaded before plugging the A200 into the PC. The A200 drivers can be downloaded, at no charge, from: *www.campbellsci.com/downloads*.

### A.1.1.2 Wiring

One end of the A200 has a terminal block while the other end has a type B female USB port. The terminal block provides 12V, G, TX, and RX terminals for connecting the GPS16X-HVS (see FIGURE A-1 and TABLE A-1).

A data cable, CSI part number 17648, ships with the A200. This cable has a USB type-A male connector that attaches to a PC's USB port, and a type B male connector that attaches to the A200's USB port.

**TABLE A-1. A200 Wiring**

| Color | Sensor Cable Label | A200 Terminal |
|---|---|---|
| Red | 12V | +12Vdc |
| Black | G | G |
| Yellow | G | G |
| White | Rx | Rx |
| Gray | Tx | Tx |
| Blue | sig ground | G |
| Shield | sig ground | G |

*FIGURE A-1. A200 Sensor-to-PC Interface*

### A.1.1.3 Powering the Sensor

The A200 provides power to the GPS16X-HVS when it is connected to a PC's USB port. An internal DC/DC converter boosts the 5 Vdc supply from the USB connection to a 12 Vdc output that is required to power the sensor.

### A.1.1.4 Determining which COM Port the A200 has been Assigned

When the A200 is loaded, the A200 is assigned a COM port number. Often, the assigned COM port will be the next port number that is free. However, if other devices have been installed in the past (some of which may no longer be plugged in), the A200 may be assigned a higher COM port number.

Often, the assigned COM port will be the next port number that is free. However, if other devices have been installed in the past (some of which may no longer be plugged in), the A200 may be assigned a higher COM port number. To check which COM port has been assigned to the A200, you can monitor the appearance of a new COM port in the list of COM ports offered in your software package (e.g., *LoggerNet*) before and after the installation, or look in the Windows Device Manager list under the ports section (access via the control panel).

## A.1.2 Using the SC110

The SC110 consists of two cables—each has a 2-foot (0.6 m) length. Use the cable that has a 9-pin female connector (DCE). TABLE A-2 shows wiring.

**TABLE A-2. SC110's DCE Cable Wiring**

| Wire Color of SC110's DCE Cable | Wire Color of GPS16X-HVS | Power Supply |
|---|---|---|
| Brown | Blue | N/A |
| White | White | N/A |
| Yellow | Shield | N/A |
| N/A | Red | +12 V |
| N/A | Black | Ground |
| N/A | Yellow | Ground |

# *Appendix B. CR23X Wiring and Programming*

## B.1 CR23X Connections

Reconfigure the receiver for 1200 baud using Garmin configuration software available on Garmin's website. Refer to TABLE B-1 to connect the GPS16X-HVS directly to a CR23X datalogger.

**TABLE B-1. CR23X Wiring**

| GPS16X-HVS | CR23X | Function |
|---|---|---|
| Red | 12V | Power In |
| Black | Ground | Power Ground |
| Yellow | Ground | Power Switch |
| White | Control Port | TXD |
| Gray | Control Port | PPS |
| Blue | Ground in operation<br>Send data to receiver for configuration | $R_x$ data |
| Shield | Ground | Shield |

## B.2 Programming

Program instruction 15 (P15) is used to read the NMEA $GPGGA string of time and position data. Each iteration of P15 can either read the numeric fields or read everything. When reading the numeric fields, such as time, latitude, longitude and elevation, P15 requires non-numeric delimiters between data points. The only available format of GPS data with delimiters is the NMEA 0183 format. Program instruction 15 (P15) reads serial data and discards non-numeric values. All non-numeric values act as delimiters between numbers, and decimal points can also act as delimiters. P15 can be used to import everything in the string, character by character, and convert it to the decimal equivalent. The decimal equivalent method is seldom used, and only when the general area (hemisphere) is not known.

### B.2.1 Program Execution Interval

When the PPS signal is used to trigger the read data function (P15), the program table execution interval does not matter. Otherwise the timing between the GPS16X-HVS output and the datalogger read must be considered. Generally the execution interval can not be less than 2 seconds when the PPS signal is not used.

## B.2.2 Reading GPS Data

TABLE B-2 is a sample CR23X P15 instruction for reading NMEA $GPGGA data string. The second parameter has two dashes indicating data buffering has been turned off. The CR10X does not have the data buffering option.

**TABLE B-2. P15 for NMEA $GPGGA Data String**

| Parameter | Data | Description |
|---|---|---|
| 1 | 1 | Repetitions |
| 2 | 66 -- | Configuration code for RS232 ASCII data at 38400 baud with data buffering turned off. The -- indicates data buffering turned off. Decimal delimiter |
| 3 | 1 | Delay before sending data out |
| 4 | 05 | Control ports. Two digit format AB. A is for handshaking and set to zero. B in this example is control port 5 (datalogger RCV). GPS16X-HVS communication cable: GPS transmit to control port 5 in this example |
| 5 | 1 | Input location where first character to transmit is stored. Note: nothing is actually transmitted |
| 6 | 0 | Number of consecutive input locations to send |
| 7 | 42 | Termination character, 42 is ASCII equivalent of the asterisk |
| 8 | 100 | Maximum number of characters to receive. |
| 9 | 80 | Delay in mS. How long to wait for $GPGGA string |
| 10 | 1 | Starting input location for time and position data |
| 11 | 1 | Multiplier, always 1. |
| 12 | 0 | Offset, always 0. |

P15 parameters 4, 5, and 10 are somewhat variable. When using a CR23X, parameter 4 can be set to 05, 06 or 07 depending on what control ports are used. Wiring of the communication cable depends on the selection for parameter 4. With a CR23X the GPS transmit wire is connected to the control port selected in parameter 4.

P15 is executed when the PPS signal drives control port 8 high. P15 will wait until one of three conditions is met: the time-out listed in parameter 9 has expired, the maximum number of characters in parameter 8 have been read, or the termination character listed in parameter 7 has been read.

P15 parameter 10 is the first input location you wish to store GPS data in. Fifteen sequential input locations will be used to store time and position.

### B.2.2.1 CR23X Example of Instruction 15 (P15)

```
Port Serial I/O (P15)
  1:  1          Reps
  2:  66 --      RS-232 ASCII (decimal delimiter), 38400 Baud
  3:  1          Delay (units = 0.01 sec)
  4:  5          Control Ports
  5:  1          Output Loc [ Bulk      ]
  6:  0          No. of Locs to Send
  7:  42         Termination Character
  8:  100        Maximum Characters
  9:  80         Time Out Delay (units = 0.01 sec)
  10: 1          Loc [ Raw_time1 ]
  11: 1          Mult
  12: 0          Offset
```

**NOTE** Communication cable wiring for:

CR23X/Example B.2.2.1 — PPS to C8, GPS transmit to C5.

## B.2.3 Filters

Filters can be used to make sure P15 reads the correct data string. Filters also ensure P15 starts to read the string at the beginning of the string. To use a filter, follow P15 with instruction P63 (extended parameters). P63 is used to define the filter. Enter the desired filter in P63.

**TABLE B-3. Filter**

| ASCII Equivalent | Character |
|---|---|
| 36 | $ |
| 71 | G |
| 80 | P |
| 71 | G |
| 71 | G |
| 65 | A |

## B.2.4 Managing the Data

Several of the data values in the $GPGGA string are too large to view or write to final storage. Some simple math is used to parse the data.

The UTC time is in the format hhmmss where hh is the hours, mm is the minutes and ss is the seconds. Six digits are too many to view with the datalogger display and some software. Add 0.3 to the raw time field. Multiply the raw time input location by 0.01 to reduce the magnitude and place the seconds in the fractional portion of the number. Next use P45 to write the integer portion (hours/minutes) to a new input location, then use P44 to write the fractional portion to another input location (seconds) and multiply that

location by 100. The last step is to use P45 again to take the integer portion of the input location for seconds. The result is hour/minutes in one input location and seconds in another.

The latitude and longitude can be parsed with the P15 instruction when decimal delimiter is on. If P15, parameter 2 is 6x, where the x selects the baud rate, every non-numeric value and decimal point will act as a delimiter. The Degrees and Minutes will be placed in one input location, and the minute fractional portion will be placed in the next input location. The decimal delimiter preserves the resolution of the original measurement.

Further parsing of the latitude and longitude may be necessary. Longitude degrees and minutes can range in value up to 18059, which exceeds the low resolution format of the dataloggers final storage area. Either parse the latitude and longitude degrees and minutes the same way the time was parsed, or store the data in high-resolution format.

The GPS quality number can be used to determine if you have a valid GPS fix and if the datalogger received the data properly. Use P89 to test if the GPS quality number is greater than or equal to one. There is a catch to using the GPS quality number to verify your data. P15 will write to fifteen input locations if everything works correctly. If P15 fails to read the GPS data, only the first input location is written to. The GPS quality number will be unchanged. If P15 fails to read the GPS data, the value displayed in the first input location will be 99999. The datalogger actually stores FFFFFFFFh, a very large number. The time field includes six digits, which can be greater than 99999. This limits the usefulness of the time field as a test for a valid GPS fix. A better approach is to overwrite the GPS quality location with zero before executing P15. Use P30 to overwrite one input location.

If the GPS time is used to set the datalogger clock, the GPS time must be parsed into three input locations: Hour, Minutes, Seconds. P114 is used to set the datalogger clock to match values in input locations. Some time will have passed between the GPS fix and when the program table reaches the P114 instruction. Adjustments can be made by adding a second or two. Be careful about setting seconds to a number greater than 59. You can also correct the UTC time to local time. Table based dataloggers require year, day, hour, minute, and seconds to use P114. Only hour, minutes, and seconds are available from the $GPGGA string. The PGRFM string includes the month, day and year, but is difficult to use.

## B.2.5 Program Discussion

Wiring when using Instruction 15:

| Function | Color | Datalogger Connection |
|---|---|---|
| Power in | Red | 12 volts |
| Power ground | Black | Ground |
| Power switch | Yellow | ground |
| TXD | White | C5 |
| PPS | Gray | C8 |
| Ground | Blue | $R_x$ data for reconfig |
| Shield | Shield | Ground |

The GPS16X-HVS needs to be reconfigured using the Garmin configuration software from the Garmin website for 1200 baud, 8 data bits, 1 stop bit and no parity (see Appendix A, *Changing GPS16X-HVS Settings (p. A-1)*). The GPGGA string should be output. The 1 pulse per second signal should be output with a pulse duration of 100 milliseconds.

The code required to read the GPS information and store it to final storage is in Subroutine 98. Subroutine 98 is interrupt driven and triggered when a rising edge is detected on Control port 8. The GPS16X-HVS has a 1 PPS signal which is wired to control port 8. The transmit data line of serial port 1 on the GPS16X-HVS is wired to control port 5. The GPS16X-HVS serial port 2 generally is not used.

When the 1 PPS signal triggers subroutine 98, P15 is executed. P15 is setup to read ASCII serial data. Each data point is separated by a non-numeric character or a decimal point. Fifteen input locations are used as temporary storage for the $GPGGA string. TABLE 4-1 explains the $GPGGA string.

The input locations used for the $GPGGA string are:

1) Raw_Time, Time in hours, minutes, and seconds
2) LatDegMin, Latitude degrees and minutes
3) Lat_Frac, Latitude fractions of minute
4) LngDegMin, Longitude degrees and minutes
5) Lng_Frac, Longitude fractions of minute
6) Quality, GPS quality indicator
7) NumSats, Number of satellites in use
8) HDPWhole, Horizontal Dilution of Precision
9) HDPFrac, Horizontal Dilution of Precision, tenths
10) Elevation, Elevation in meters
11) Geoidal, Geoidal separation in meters
12) Geoidalth, Geoidal separation in meters, tenths
13) Age, Age of differential GPS data
14) Agetenth, Age of differential GPS data, tenths
15) DiffID, Differential reference station ID

Additional input locations used in the example program are:

18) Orig_TM, Copy of original time
19) Int1, Place holder for math
20) Hours, formatted hours
21) Minutes, formatted minutes
22) Seconds, formatted seconds
23) remainder, place holder for math

Before writing any datalogger code, it's best to enter all the input locations needed. In Edlog, open the input location editor (F5) and enter names for the input locations listed above. When an input location is needed, use the input location pick list (F6).

### B.2.5.1 CR23X Example Program

```
;{CR23X}
;

*Table 1 Program
  01:  60         Execution Interval (seconds)

; Instruction to eliminate warning about unused subroutine, not needed
1:  If Flag/Port (P91)
  1:   11         Do if Flag 1 is High
  2:   98         Call Subroutine 98

*Table 2 Program
  02:  0.0000     Execution Interval (seconds)

*Table 3 Subroutines

1:  Beginning of Subroutine (P85)
  1:   98         Subroutine 98

;--- read serial data non-buffered
2:  Port Serial I/O (P15)
  1:   1          Reps
  2:   66 --      RS-232 ASCII (decimal delimiter), 38400 Baud
  3:   1          Delay (0.01 sec units) before TX
  4:   5          No RTS/DTR, C5 TXD/RXD
  5:   1          Start Loc for TX [ Raw_Time  ]
  6:   0          Number of Locs to TX
  7:   42         Termination Character for RX
  8:   100        RX Buffer Size or Max Chars to RX if Par 2 indexed (--)
  9:   80         Time Out for CTS (TX) and/or RX (0.01 sec units)
  10:  1          Start Loc for RX [ Raw_Time  ]
  11:  1.0        Mult for RX
  12:  0.0        Offset for RX

;--- filter for $GPGGA
3:  Extended Parameters (P63)
  1:   36         Option ;$
  2:   71         Option ;G
  3:   80         Option ;P
  4:   71         Option ;G
  5:   71         Option ;G
  6:   65         Option ;A
  7:   0          Option
  8:   0          Option

; Test for valid GPS fix and string read
4:  If (X<=>F) (P89)
  1:   6          X Loc [ Quality   ]
  2:   3          >=
  3:   1          F
  4:   30         Then Do

; Make a copy of time
5:  Z=X (P31)
  1:   1          X Loc [ Raw_Time  ]
  2:   18         Z Loc [ Orig_TM   ]

; Add 0.45 to time stamp to eliminate complications with
; floating point math, P44, and P45
6:  Z=X+F (P34)
  1:   18         X Loc [ Orig_TM   ]
  2:   0.45       F
  3:   18         Z Loc [ Orig_TM   ]
```

```
; Move minutes and seconds right of decimal
7:  Z=X*F (P37)
  1:   18         X Loc [ Orig_TM   ]
  2:   .0001      F
  3:   19         Z Loc [ Int1      ]

; Pluck off hours
8:  Z=INT(X) (P45)
  1:   19         X Loc [ Int1      ]
  2:   20         Z Loc [ Hours     ]

; Subtract hours out
9:  Z=X-Y (P35)
  1:   19         X Loc [ Int1      ]
  2:   20         Y Loc [ Hours     ]
  3:   19         Z Loc [ Int1      ]

; Move decimal left 2 places
10:  Z=X*F (P37)
  1:   19         X Loc [ Int1      ]
  2:   100        F
  3:   19         Z Loc [ Int1      ]

; Pluck off minutes
11:  Z=INT(X) (P45)
  1:   19         X Loc [ Int1      ]
  2:   21         Z Loc [ Minutes   ]

; Subtract out minutes
12:  Z=X-Y (P35)
  1:   19         X Loc [ Int1      ]
  2:   21         Y Loc [ Minutes   ]
  3:   19         Z Loc [ Int1      ]

; Move decimal left 2 places
13:  Z=X*F (P37)
  1:   19         X Loc [ Int1      ]
  2:   100        F
  3:   19         Z Loc [ Int1      ]

; Pluck of seconds
14:  Z=INT(X) (P45)
  1:   19         X Loc [ Int1      ]
  2:   22         Z Loc [ Seconds   ]

; Write data to final storage every time there is
; a valid read of GPS data
15:  Do (P86)
  1:   10         Set Output Flag High (Flag 0)

16:  Set Active Storage Area (P80)^18796
  1:   1          Final Storage Area 1
  2:   101        Array ID

; Write datalogger based time stamp
17:  Real Time (P77)   ^27570
  1:   0011       Hour/Minute,Seconds (midnight = 0000)

; Write GPS based time stamp
18:  Sample (P70)   ^6080
  1:   3          Reps
  2:   20         Loc [ Hours     ]

; Set resolution to high for latitude and Longitude
19:  Resolution (P78)
  1:   1          High Resolution
```

```
20:  Sample (P70)   ^20303
  1:   4        Reps
  2:   2          Loc [ LatDegMin ]

; Write elevation in meters
21:  Sample (P70)   ^32246
  1:   1          Reps
  2:   10         Loc [ Elevation ]

; Set resolution low
22:  Resolution (P78)
  1:   0          Low Resolution

; Write the number of satellites in view
23:  Sample (P70)   ^1910
  1:   1          Reps
  2:   7          Loc [ NumSats   ]

; Reset the the GPS quality number
24:  Z=F x 10^n (P30)
  1:   -1         F
  2:   00         n, Exponent of 10
  3:   6          Z Loc [ Quality   ]

25:  End (P95)

26:  End (P95)

End Program
```

# B.3 Troubleshooting

The first step is to verify the GPS16X outputs with a terminal program (see Section 6.1, *Testing and Evaluating Serial Communications (p. 13)*). The second step is to verify that it really does not work. With the GPS16X-HVS running and the datalogger program running, look at the input location for GPS Quality Number. This location will show a one when the GPS16X-HVS output is picked up by the datalogger. The input location for parsed time and position are good locations to check. The location for seconds should update every time the GPS data is updated.

If the GPS time and position data are not shown in the input locations, check the communication cable wiring.

If the GPS16X-HVS data is not correct every program table execution but correct sometimes, check the P15 time-out. It may need a longer time-out. Also check the P15 maximum number of characters to receive, usually 100 is enough. Check the P15 termination character; it should be set to 42 (*). The termination character should also work if set to 13 or 10. Also check the buffering and filter. Buffering should be turned off by indexing parameter 2.

For P15 to properly read the $GPGGA string, P15 must be executing while the $GPGGA string starts and finishes. The P15 time-out needs to be long enough to pick up the string. The string is output once a second. If P15 starts to execute while the GPS16X-HVS is sending the string, P15 must wait until the string is sent again plus the amount of time it takes to send the string. It shouldn't need more than 1.5 seconds. P15 time-out is in units of 0.01 seconds, 100 = 1 second. A longer time-out will force the datalogger to wait until the time-out has expired or the termination character is received or the

maximum number of characters are received. If the data in input locations seem to move from the proper input location to another input location, P15 is stopping before the entire string has been read. An example is latitude being displayed in the time field, then in the latitude field. P15 works best when P15 quits reading data because the termination character has been read. Using the PPS to trigger subroutine 98 is the best way to start P15 just before the GPS16X-HVS sends the $GPGGA string. If the PPS signal pulls C8 high while the datalogger is in the middle of executing an instruction, it may not be able to run subroutine 98 before the $GPGGA string has started, which will cause the datalogger to miss the data string. Turning on the data buffering may remedy the problem. Lengthening the serial time-out to allow P15 to execute for 2 cycles of NMEA output may help. Otherwise the SDM-SIO4 may be required or the datalogger program will need to be simplified.

The datalogger will not pick up valid data until the GPS16X-HVS has a valid GPS fix, except during a GPS16X-HVS warm start where time can be read before position is known. Don't spend a lot of time trouble shooting a phantom problem just because the GPS receiver does not have a valid GPS fix.

# Appendix C. Using an SDM-SIO4

## C.1 SDM-SIO4 Connections

An SDM-SIO4 should be used if measuring the GPS16X-HVS with a CR5000 or CR9000X datalogger. An SC110 cable is required to connect the GPS16X-HVS to an SDM-SIO4. The SC110 consists of two cables. Use the cable that terminates in a 9-pin male connector (DTE). The stripped and tinned leads of that cable attach to the GPS16X-HVS (see TABLE C-1).

**TABLE C-1. SC110's Cable Wiring**

| Wire Color of SC110's Cable | Wire Color of GPS16X-HVS | Power Supply |
|---|---|---|
| Brown | Gray | N/A |
| White | White | N/A |
| Yellow | Shield | N/A |
| N/A | Red | +12 V |
| N/A | Black | Power Ground |

## C.2 CR9000X Example Program

```
'NMEAGGA_Sio4_030805MGW1.CR9
'This program acquires NMEA GGA data from a GPS receiver using the SDM-SIO4.
'_____
'Notes:
'(1) Data is acquired from NMEA0183 $GPGGA string:
'    Sio4Fields:  GGAFields:      Definitions:
'    f1           GGA(Field1)     GGA_UTC_Time of position
'    f2,f3        GGA(Field2)     Latitude
'    f4,          GGA(Field3)     North or South indication letter
'    f5,f6        GGA(Field4)     Longitude
'    f7           GGA(Field5)     East or West indication letter
'    f8           GGA(Field6)     GPS quality,0=NoGPS,1=GPS,2DGPS
'    f9           GGA(Field7)     Number of satellites in use
'    f10          GGA(Field8)     HDOP, Horizontal Dilution Of Precision
'    f11          GGA(Field9)     Antenna altitude in Meters
'                 GGA(Field10)
'                 GGA(Field11)    Geoidal separation in Meters
'                 GGA(Field12)
'                 GGA(Field13)    Age of differential GPS data
'                 GGA(Field14)    Differential reference station
'(2) SIO4 programming:
'    fltst 1 "t[$GPGGA,]xFt[,]Dt[.]Dt[,]b1t[,]Dt[.]Dt[,]b1t[,]Ft[,]Ft[,]Ft[,]FX"
'_
 Const OneRep=1
 Const NoValues=0
 Const OneValue=1
 Const ElevenGGAValues=11
'..
 Const UnityMultiplier=1.0
 Const NoOffset=0.0
'..
 Const Sio4Address0=0
 Const Port2=2
```

```
'..
'SDM-Sio4 command codes:
 Const UnusedParameter = 0000
 Const PollForData0001 = 0001
 Const SendDataToLgr   = 0004
 Const Sio4COMSetUpCmd = 2049
 Const StartRxFilter   = 2054
 Const Port2ComCode    = 9147 '9=NoHandshaking; 1=1StopBitNoParity; 4=8DataBits; 7=19200Baud
 Const RxFilt9001      = 9001 'Command parameter for user defined fltst #1.
 Dim   DataPoll,NotUsed
'..
 Public RawGGAData(ElevenGGAValues)
 Alias  RawGGAData(1)=GGA_UTC_Time
 Alias  RawGGAData(2)=Latt_Int    : Units Latt_Int=Deg
 Alias  RawGGAData(3)=Latt_Frac   : Units Latt_Frac=Deg
 Alias  RawGGAData(4)=LattH_NS
 Alias  RawGGAData(5)=Longit_Int  : Units Longit_Int=Deg
 Alias  RawGGAData(6)=Longit_Frac : Units Longit_Int=Deg
 Alias  RawGGAData(7)=LongH_EW
 Alias  RawGGAData(8)=GPSQuality
 Alias  RawGGAData(9)=Satellites
 Alias  RawGGAData(10)=HDOP
 Alias  RawGGAData(11)=Altitude   : Units Altitude=Meters
'_
 DataTable(GPSData,True,-1)
 DataInterval(0,0,0,0)
 Sample(ElevenGGAValues,RawGGAData(),IEEE4)
 EndTable
'________
 BeginProg
  '.....................................................
  'Configure SDM-Sio4 Port#2 for communications with GPS port:

SDMSIO4(NotUsed,OneRep,Sio4Address0,Port2,Sio4COMSetUpCmd,Port2ComCode,UnusedParameter,NoValues
,UnityMultiplier,NoOffset)
   Delay(100,mSec)
  '.....................................
  'Start GGA data filter on SDM-Sio4 port:

SDMSIO4(NotUsed,OneRep,Sio4Address0,Port2,StartRxFilter,RxFilt9001,UnusedParameter,NoValues,Uni
tyMultiplier,NoOffset)
   Delay(20,mSec)
  '_____________________________
   Scan(50,mSec,0,0) 'Main Scan:
    '..

SDMSIO4(DataPoll,OneRep,Sio4Address0,Port2,PollForData0001,UnusedParameter,UnusedParameter,OneV
alue,UnityMultiplier,NoOffset)
     If DataPoll>0 Then
       Delay(10,mSec)

SDMSIO4(RawGGAData(),OneRep,Sio4Address0,Port2,SendDataToLgr,UnusedParameter,UnusedParameter,El
evenGGAValues,UnityMultiplier,NoOffset)
       Delay(10,mSec)
       CallTable(GPSData)
     EndIf
    '..
   NextScan
'_______
 EndProg
```

## Campbell Scientific Companies


**Campbell Scientific, Inc.**
815 West 1800 North
Logan, Utah 84321
UNITED STATES
*www.campbellsci.com* • info@campbellsci.com

**Campbell Scientific Canada Corp.**
14532 – 131 Avenue NW
Edmonton AB T5L 4X4
CANADA
*www.campbellsci.ca* • dataloggers@campbellsci.ca

**Campbell Scientific Africa Pty. Ltd.**
PO Box 2450
Somerset West 7129
SOUTH AFRICA
*www.campbellsci.co.za* • cleroux@csafrica.co.za

**Campbell Scientific Centro Caribe S.A.**
300 N Cementerio, Edificio Breller
Santo Domingo, Heredia 40305
COSTA RICA
*www.campbellsci.cc* • info@campbellsci.cc

**Campbell Scientific Southeast Asia Co., Ltd.**
877/22 Nirvana@Work, Rama 9 Road
Suan Luang Subdistrict, Suan Luang District
Bangkok 10250
THAILAND
*www.campbellsci.asia* • info@campbellsci.asia

**Campbell Scientific Ltd.**
Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM
*www.campbellsci.co.uk* • sales@campbellsci.co.uk

**Campbell Scientific Australia Pty. Ltd.**
PO Box 8108
Garbutt Post Shop QLD 4814
AUSTRALIA
*www.campbellsci.com.au* • info@campbellsci.com.au

**Campbell Scientific Ltd.**
3 Avenue de la Division Leclerc
92160 ANTONY
FRANCE
*www.campbellsci.fr* • info@campbellsci.fr

**Campbell Scientific (Beijing) Co., Ltd.**
8B16, Floor 8 Tower B, Hanwei Plaza
7 Guanghua Road
Chaoyang, Beijing 100004
P.R. CHINA
*www.campbellsci.com* • info@campbellsci.com.cn

**Campbell Scientific Ltda.**
Fahrenheitstraße 13
28359 Bremen
GERMANY
*www.campbellsci.de* • info@campbellsci.de

**Campbell Scientific do Brasil Ltda.**
Rua Apinagés, nbr. 2018 ─ Perdizes
CEP: 01258-00 ─ São Paulo ─ SP
BRASIL
*www.campbellsci.com.br* • vendas@campbellsci.com.br

**Campbell Scientific Spain, S. L.**
Avda. Pompeu Fabra 7-9, local 1
08024 Barcelona
SPAIN
*www.campbellsci.es* • info@campbellsci.es


*Please visit www.campbellsci.com to obtain contact information for your local US or international representative.*